



AF
Ifw

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on:

Date: April 23, 2007

By: _____

Susan L. Baka

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: PUJARE, *ET AL.*

APPLICATION No.: 09/826,607

FILED: APRIL 5, 2001

FOR: CONVENTIONALLY CODED APPLICATION
CONVERSION SYSTEM FOR STREAMED
DELIVERY AND EXECUTION

EXAMINER: SHIN, K.H.

ART UNIT: 2143

CONF. NO: 4038

Transmittal for Appellant's Reply Brief

**Mail Stop Appeal Briefs-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450**

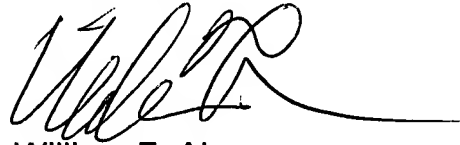
Sir:

In response to the Examiner's Answer dated February 23, 2007, Applicant submits the enclosed as follows:

- ☒ Appellant's Reply Brief
- ☒ Form PTO/SB/32 Request for Oral Hearing Before the Board of Patent Appeals and Interferences
- ☒ Fee For Request for Oral Hearing (37 C.F.R. § 41.20(b)(3)):
 - ☐ Small Entity: \$500.00 ☒ Large Entity: \$1000.00
 - ☒ Enclosed is a check for \$1,000.00.
 - ☒ Please charge any underpayment in fees for timely consideration of this transmittal and enclosures to Deposit Account No. 50-2207.

- ☒ Applicant petitions for an Extension of Time if necessary for timely filing of this Brief.

Respectfully submitted,
Perkins Coie LLP



William F. Ahmann
Registration No. 52,548

Date: April 23, 2007

Correspondence Address:

Customer No. 22918
Perkins Coie LLP
P. O. Box 2168
Menlo Park, California 94026-2168
(650) 838-4300



Attorney Docket No. 30126-8013.US01

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF:

Pujare *et al.*

APPLICATION NO.: 09/826,607

FILED: APRIL 5, 2001

**FOR: CONVENTIONALLY CODED APPLICATION
CONVERSION SYSTEM FOR STREAMED
DELIVERY AND EXECUTION**

EXAMINER: Kyung H. Shin

ART UNIT: 2143

CONFIRMATION NO: 4038

APPELLANT'S REPLY BRIEF

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Mail Stop Appeal Brief - Patents

Sir:

This is a Reply Brief filed in response to the Examiner's Answer mailed February 23, 2007. This Reply Brief is intended to supplement, but not replace, the Brief on Appeal filed with the Board of Patent Appeals and Interferences from the decision of Examiner Shin mailed June 19, 2006, in which pending claims 1-52 stand in final rejection.

The present paper is Appellant's Reply Brief submitted in compliance with 37 C.F.R. §41.41.

ARGUMENT

I. Independent Claim 1

On page 4 of the Examiner's Answer, the Examiner asserts:

Regarding Claims 1, 14, 27, 40, Eylon discloses a process, apparatus, method for converting a conventionally coded computer application program into a data set suitable for streamed delivery across a network from a server and concurrent execution on a client in a computer environment, comprising the steps of:

- a) providing installation monitoring means for monitoring an installation process of the conventionally coded application program on a local computer system; (see Eylon col. 3, lines 45-50: server; col. 3, lines 52-56; col. 4, lines 51-56: streamed application ; col. 8, lines 49-53: monitor and management, streamed application installation on local system)

Eylon discloses wherein the installation monitoring means gathers modification information (see Eylon col. 8, lines 49-53: application manager monitors installation process; col. 7, lines 52-55: database for storage of gathered information), and providing data set creation means for processing the modification information for converting the application program into a data set suitable for streaming bits of the data set over the network to the client (see Eylon col. 3, lines 52-56; col. 4, lines 51-56: streamed application) such that the application program is capable of beginning execution on the client prior to downloading all of the application program (see Eylon col. 3, lines 52-56: initiate execution after fraction of application loaded(i.e. before entire application downloaded))

The assertion by the Examiner is that Eylon **converts** a conventionally coded application program **into a data set suitable for streamed delivery** across a network **from a server**... The language of this assertion is significant because, as will be shown, Eylon sends "streamlets" (i.e., portions of a streamed application) from a server to a client and an application manager at the client monitors usage of the program. There is no conversion process at the client. Yet, the Examiner uses the application manager to teach gathering modification information that is used to convert a conventionally coded application into a streamed application.

With specific reference to the Examiner's assertions and support therefore, the Examiner asserts "Eylon discloses wherein the installation monitoring means gathers modification information (see Eylon col. 8, lines 49-53: application manager monitors installation process; col. 7, lines 52-55: database for storage of gathered information)[.]" Notably, at col. 8, lines 49-53, Eylon describes "The Application Manager 110."

Eylon clearly describes the application manager 110 as part of the client system 14. Specifically, at col. 8, lines 1-4, "FIG. 4 is an illustration of a more detailed block diagram of **the client system 14** showing various streaming control modules which comprise a preferred implementation of **the streaming support system 102**." And at col. 8, lines 39-41, "In addition to the VFS 160 and FSD driver 150, **the streaming support system 102** can also comprise **an application manager 110** and a communication driver 170." Therefore, the application manager 110 is on the client system 14, which receives streamlets from a server (i.e., the client system executes a streamed application, which the application manager 110 monitors).

The second citation the Examiner uses to support the assertion is at col. 7, lines 52-55, which reads, "A user database 173 can also be provided for storing various user-specific information for use in authorizing access to streamed application, storing information related to streaming order, etc." Notably, the user database 173 is stored at the server. Moreover, the "authorizing access" functionality of the database ensures

that the client will not receive a streamed application unless and until the client is authorized. It follows that the monitoring accomplished by the application manager 110 at the client will not be of any use in this regard. That is, the client has to be authorized to access a streamed application before the client will be able to monitor the streamed application. The "information related to streaming order," on the other hand, is actually used by the server in order to stream the streamed application to the client. It follows that this information will be available prior to attempting to stream to the client. Therefore, the Examiner's implication that the information gathered by the application monitor is stored in a database associated with authorizing access to a streamed application and the order in which streamlets are sent to a client is without support in Eylon and is, in fact, nonsensical.

With specific reference to the Examiner's assertions and support therefore, the Examiner asserts, "providing data set creation means for processing the modification information for converting the application program into a data set suitable for streaming bits of the data set over the network to the client (see Eylon col. 3, lines 52-56; col. 4, lines 51-56: streamed application)[.]"

In support of this assertion, the Examiner refers to the Summary of the Invention at col. 3, lines 52-56, which reads, "the application is streamed to the client's system in streamlets or blocks which are stored in a persistent client-side cache and the system is configured such that the application can begin to execute on the client machine after only a small fraction of the application is loaded." The Examiner also refers to the Summary of the Invention at col. 4, lines 51-56, which reads, "the present architecture described enables a positive user experience with streamed applications without requiring constant use of broadband data links. In addition, and unlike remote-access application systems, the streaming applications execute on the client machine and server and network resources are utilized for delivery only."

Notably absent from the Examiner's citations are any description regarding processing the modification information (i.e., information gathered from monitoring an installation process of a conventionally coded application program on a local computer system). Therefore, the citations fail to even address the claim language that the Examiner implies they do. Specifically, there is nothing analogous to a "means for processing the modification information" in the citations or, indeed, anywhere in Eylon.

With specific reference to the Examiner's assertions and support therefore, the Examiner asserts, "such that the application program is capable of beginning execution on the client prior to downloading all of the application program (see Eylon col. 3, lines 52-56: initiate execution after fraction of application loaded (i.e. before entire application downloaded))." This assertion simply reinforces the applicants' point that the application program is streamed to the client. This directly contradicts the Examiner's implication that the application monitor gathers information used to convert a conventionally coded application to a streamed application.

The Examiner admits at page 4 of the Examiner's Answer that "Eylon does not specifically disclose the capability of redirecting registry information thereby creating a registry spoofer capability of the capability for the parameterization of configuration data." The Examiner uses Schmeidler to make up for this deficiency. However, Schmeidler is similar to Eylon in that an application is streamed to a client. So, the data gathered at the client would not be used to convert a conventionally coded application into a streamed application.

The Examiner relies upon Kumar and Schmeidler to disclose "parameterizing registry modifications...." Kumar is not even in the field of streaming software.

Accordingly, like Eylon and Schmeidler, Kumar does not have any teachings related to the conversion of a conventionally coded application into a stream-enabled application.

II. Dependent Claims 2-13

On page 6 of the Examiner's Answer, the Examiner asserts Eylon discloses various aspects of the dependent claims. The applicants respond briefly simply to point out that Eylon does not disclose "data set creation means" as claimed in claims 2, 3, 5, and/or 6 because Eylon does not disclose the conversion of a conventionally coded application into a stream-enabled application. Rather, Eylon discloses the execution of a stream-enabled application at a client.

Eylon does not disclose "a user interface that allows an operator to examine all changes made to said local computer system during said installation process and to edit said modification information," as claimed in claim 7. This is true at least because Eylon does not disclose an installation process, but also because Eylon cannot edit modification information even with respect to the streamed application. Indeed, if Eylon were modified to include teachings related to modification of information (e.g., using Kumar, Schmeidler, and/or Cheng), the user would be unable to accomplish anything meaningful; the server provides the data necessary to execute the stream-enabled process locally, and the streamlets themselves as appropriate. A user at the local client would not know what changes could be made. The Examiner has provided no suggestion as to what a user at the local client could enter that would be of any value, or how a user could efficiently enable an automatic update, etc. by entering data in a user interface at the local client.

On page 8 of the Examiner's Answer, with reference to claim 8, the Examiner asserts Eylon discloses "the installation monitoring means monitors the application as it runs (see Eylon col. 8, lines 49-53: application manager monitors installation process)." As has been stated previously, Eylon does not disclose an installation monitoring means (and never refers to the streaming application monitor as an installation monitor).

In the words of Eylon at col. 8, lines 41-44, "The application manager is preferably configured to initiate execution of a streaming application after sufficient streamlets have been received." Notably, Eylon states that the streamlets are received, not installed, and the application manager monitors usage of the program, not the installation process.

On page 9 of the Examiner's Answer, with reference to claim 13, the Examiner asserts Eylon discloses "the installation monitoring means records a state of the local computer system before the installation process begins to give a more accurate picture of any modifications that are observed by the installation monitoring means. (see Eylon col. 7, lines 52-55: database for application configuration data, installation file information stored such that setup can be duplicated on multiple machines.)" As previously stated, the installation monitoring means of claim 1, from which claim 13 depends, gathers information from installation of a conventionally coded application program that is used to stream the application ("a data set suitable for deceiving said client into allowing streaming of bits of said data set over said network to said client such that said application program is capable of beginning execution on said client prior to downloading all of said application program"). Eylon, at col. 7, lines 52-55, describes a database used to grant authorization to a stream-enabled application or information related to streaming order. Therefore, the database of Eylon must be used prior to what the Examiner refers to as "the installation process" at the client. (Actually, it is not an installation process, and Eylon would not refer to it as such. The appellants are simply using the language of the Examiner in this case.)

With specific reference to claim 13, Eylon does not disclose "said installation monitoring means records a state of said local computer system before said installation process begins to give a more accurate picture of any modifications that are observed by said installation monitoring means." While Eylon does not disclose an installation process at all, the appellants note that the application monitor of Eylon does not even record state of the local computer system before executing the stream-enabled

application. Moreover, the Examiner does not address the issue of what modifications may be made. Eylon makes no specific reference to a change of state at the local client.

III. Claims 14-52

While claims 14-52 may have scopes that are different from those of claims 1-13, claims 14-52 are allowable for reasons at least similar to those of claims 1-13.

IV. Response to Argument

Starting at page 11 of the Examiner's Answer, the Examiner provides a response to appellant's previous arguments.

1. Eylon is not prior art

The Examiner argues that Eylon is 102(a) prior art. That is, "... the invention was known in this country ..., before the invention thereof by the applicant for patent..." However, the Examiner is relying upon a provisional filing date, not the filing date of the Eylon reference. The contents of the Eylon provisional application could be entirely different from the Eylon patent as issued. The appellants respectfully assert that the Eylon reference itself is not 102(a) prior art, and if the Examiner wishes to cite actual 102(a) prior art, such as the Eylon provisional application (if applicable), the Examiner should do so.

2. Kumar is not prior art

The Examiner argues that Kumar is within the field of endeavor for delivery of software. By this statement, the Examiner has underscored his fundamental lack of understanding in this case. The claims are not directed to the delivery of software, but rather to the conversion of a conventionally coded application to a stream-enabled application.

Moreover, the appellants take issue with the characterization of "Applicant's Invention" as "the delivery and installation of software over network communications." The claims are directed to the installation of a conventionally coded application. The delivery of software to the local machine is not claimed, or even described, because the means by which the conventionally coded application becomes available to the local machine is irrelevant. Installation of a conventionally coded application is one part of converting the conventionally coded application to a stream-enabled application that is, of course, capable of being streamed.

3. The conversion of a conventionally coded application

The Examiner argues that Kumar does not convert a conventionally coded application for streamed software delivery, but Eylon does. The Examiner supports this assertion by describing streaming as a delivery mechanism, then: "The Eylon prior art discloses the capability to format an application for streamed delivery, which is equivalent to applicant's invention. (see Eylon col. 5, lines 53-64). The Eylon prior art does not disclose or mentions a stream-enabled application stored within any system. The Eylon prior art prepares the application for stream delivery by the generation of streamlets." The appellants note that this is a new argument on the part of the Examiner, specifically, that Eylon discloses a technique that is equivalent to that described in the claims.

At col. 5, lines 53-64, Eylon describes preparation for streaming. Specifically: "Prior to streaming an application, the application files are divided into small segments called streamlets. Rather than delivering an entire application prior to execution, the server delivers information about the application files and preferably a small portion of the application itself. In particular, the client receives a file structure specification which defines how files associated with the application and required for the application to operate appear to a computer when the application is locally installed. In particular, the file structure specification defines the structure of at least the primary application file which is loaded by the operating system when the application is initially executed."

As an initial matter, the appellants note that Examiner has used the term "installed" in a manner that is contrary to that used by Eylon. That is, Eylon describes a file structure specification for use when an application is streamed to a local client so that the local client knows what the application would look like if it were locally installed (instead of streamed). The Examiner, on the other hand, asserts that the streamed application is itself installed on the client.

To render a claim obvious, the prior art must teach each and every element of the claim. The Examiner has failed to show that Eylon teaches each and every element of the claims, but now asserts that the entire claim is rejected because "The Eylon prior art discloses the capability to format an application for streamed delivery, which is equivalent to applicant's invention. (see Eylon col. 5, lines 53-64)." The appellants respectfully assert that equivalence is not a basis for the rejection of a claim. It is possible to patent a first process that produces a result that is the same as a second process, so long as the steps of the first process are new, useful, and non-obvious. Eylon apparently breaks an application into streamlets to be sent to a client. The appellant's claims, on the other hand, describe a new, useful, and non-obvious technique for converting a conventionally coded application into a stream-enabled application. The result of the claimed technique may even be superior to the result of the Eylon technique. Nevertheless, it is the way in which the stream-enabled application is produced that is novel in this case, not the stream-enable application itself or the degree of optimization of the result.

Moreover, streaming software is a relatively new field (different from streaming media). In the year 2000 and thereabout, streaming techniques were being developed and modified for the first time. It is not a given that Eylon had a complete understanding of how to break an application into streamlets in an optimum manner. Their technique involves breaking the application files into streamlets "corresponding generally to various portions of the application files... File loads issued by the operating system to the virtual file system are translated to determine which streamlets correspond to the

load request and the appropriate data is returned." (Eylon Abstract). The appellants believe this text means Eylon broke files into multiple pieces. Each piece is associated with a file. When the operating system wants the file, the virtual file system provides all of the streamlets associated with the requested file. The appellant's claimed techniques may be superior in that the stream-enabled application includes better data, such as registry modifications.

At pages 13-14 of the Examiner's Answer, the Examiner asserts "Registry configuration parameters must be setup and installed (i.e., some form of an installation) on a client system in order to execute even a streamed application. The Eylon prior art discloses the execution of a streamed application on a client system. In addition, the Eylon prior art discloses the capability to monitor application installation and processing. (see Eylon col. 8, lines 49-53)." As has previously been mentioned, Eylon does not refer to the provisioning of a portion of an application to a client as the installation of the application on the client. The appellants also do not refer to the provisioning of a portion of an application to a client as the installation of the application on the client.

Moreover, claim 1 includes the language "said installation monitoring means gathers modification information including system registry modifications that said installation process makes to certain file paths in a system registry of said local computer system." While the changes are made to a system registry of a local computer when an application is installed, the appellants believe no such changes are made to a system registry of a client when the client executes a streamed application. Registry modifications are made upon installation, not upon streaming.

The remainder of the Examiner's Answer is largely irrelevant, as it has to do with the delivery of the stream-enabled application. The claims are directed to the

conversion of a conventionally coded application to a stream-enabled application. However, the appellants specifically take issue with the text: "Applicant's invention has two principal features: a) streamed transfer and execution of an application file, and b) capability for monitoring and storage of installation information."

The claims are not directed to a streamed transfer. Also, the Examiner refuses to recognize the difference between a conventionally coded application and a stream-enabled application. A difference is drawn between non-stream-enabled applications and stream-enabled applications in all prior art that has anything to do with streaming software and, more importantly, in the appellant's specification. Only the Examiner fails to recognize the significance, as evidenced by the fact that the Examiner states a principal feature is "execution of an application file" without stating whether he is referring to a stream-enabled application or a non-stream-enabled application. Nevertheless, the claims are not specifically directed to execution of an application file, but rather the conversion of a conventionally coded application to a stream-enabled application. The executability of the stream-enabled application is a desired result of the claimed technique.

With respect to the second "principal feature," the Examiner refers to capability for monitoring and storage of installation information. However, the claims are not simply directed to monitoring an installation; the information derived from the monitoring is both significant and claimed. The Examiner does not acknowledge the importance of gathering monitoring information that is used to create a stream-enabled application, as claimed. This is despite the fact that the appellants have pointed out the significance and the claim language on multiple occasions.

4. After final remarks

The appellants have successfully responded to all issues addressed by the Examiner and pointed out clearly why the Examiner was wrong. Accordingly, a response to the Examiner's remaining points is deemed duplicative.

In view of the foregoing remarks, Appellants submit that the pending claims are in condition for allowance and patentably define over the prior art, and urge the Board to overturn the Examiner's rejections.

Respectfully submitted,
Perkins Coie LLP

A handwritten signature in black ink, appearing to read 'William F. Ahmann', with a long horizontal flourish extending to the right.

William F. Ahmann
Registration No. 52,548

Date: April 23, 2007

Correspondence Address:

Customer No. 22918
Perkins Coie LLP
P. O. Box 2168
Menlo Park, California 94026-2168
(650) 838-4300